BlueGene/L Consortium

System Software Workshop

Pete Beckman

Feb 23 – 24, Salt Lake City, Utah

Agenda Notes: Wed

- 08:00 Welcome and Introduction
- 08:30 Tech Session 1
- 10:30 Break
- 11:00 Tech Session 2
- 12:30 Lunch on your own
- 01:30 Tech Session 3
- 03:30 Break
- 03:45 Tech Session 4
- 05:00 Dismiss
- 06:00 Reception
- 08:00 Dinner on your own

Agenda Notes: Thurs

- 08:30 Tech Session 1
- 10:30 Break
- 11:00 Tech Session 2
- 12:30 Lunch on your own
- 01:30 Tech Session 3
  - Futures Round Table
  - BG Consortium, Purchasing, Access
- 03:00 Dismiss

Special Thanks

Goals for Workshop

- Learn about the system software currently available and in development for BG/L
- Share experiences and solutions
- Prioritize system software enhancements and requirements
- Find areas where community Open Source development can extend the BG/L environment
- Organize community efforts and build collaboration

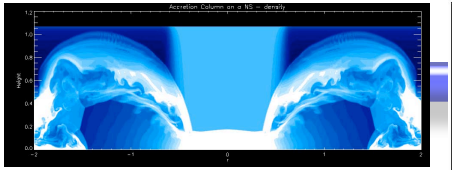
The Road To Petaflops

- Petaflop computing is just a couple years away, and BlueGene-like architectures will lead the way
- What **didn't** we do to get here?
  - Develop new exotic automatically parallelizing compilers
  - Develop new multi-threaded functional programming languages to express more parallelism
  - Improve debugging

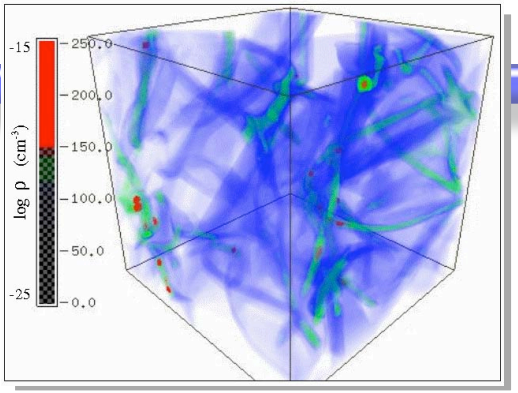
What Did We Do:

- Standardized on a single Open Source development environment: Linux
- Created Open Source scalable libraries and performance tools
- Designed scalable, parallel I/O semantics
- Developed sophisticated, multi-component application frameworks
- Made programming environments **more** rich and complex, adding perl, python, and dynamically loaded libraries
- Addressed power consumption and density
- *Refactored system architecture and software*
- Complained about debugging

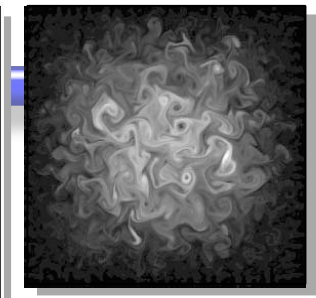
The Petaflop Applications Have Already Been Written



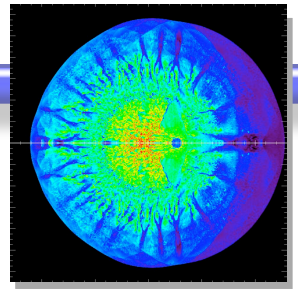
Shortly: Relativistic accretion onto NS



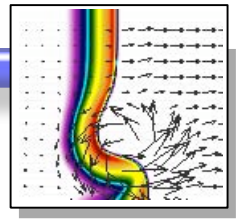
Gravitational collapse/Jeans instability



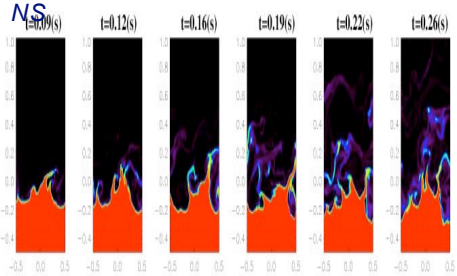
Compressed turbulence



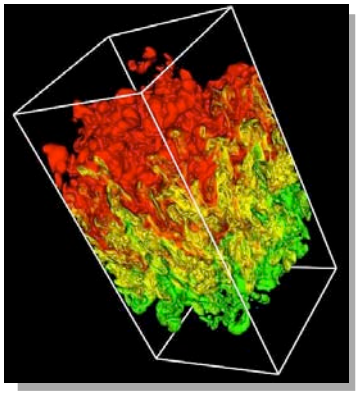
Type Ia Supernova



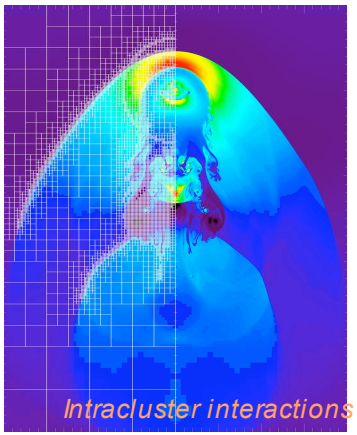
Flame-vortex interactions



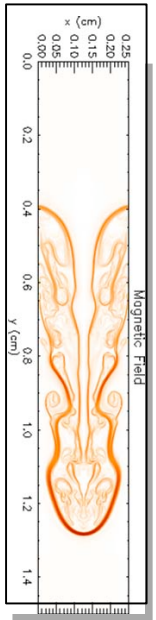
Wave breaking on white dwarfs



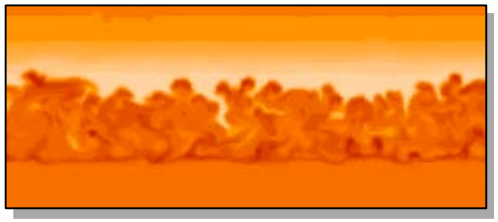
Rayleigh-Taylor instability



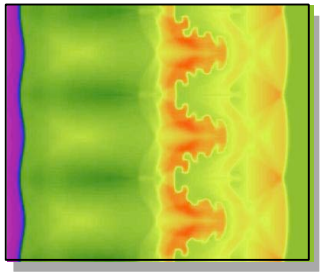
Intracluster interactions



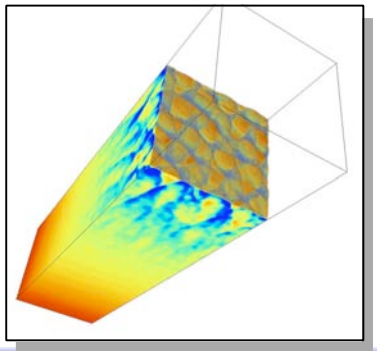
Magnetic Rayleigh-Taylor



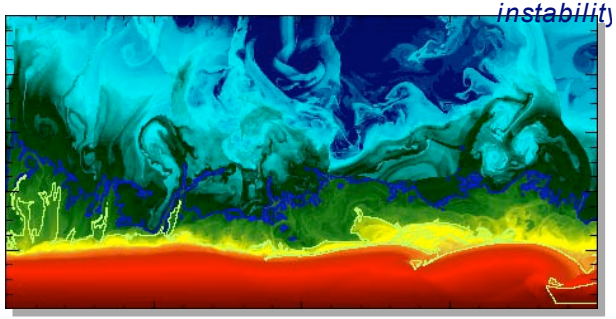
Nova outbursts on white dwarfs



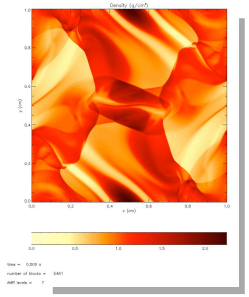
Laser-driven shock instabilities



Cellular detonation



Helium burning on neutron stars



Orzag/Tang MHD vortex

Richtmyer-Meshkov instability

We Have Two Challenges

- Scale existing applications on BG/L. Some will be petaflop candidates
  - Improve floating point code
  - Replace bottlenecks with scalable I/O, data distributions, and algorithms
- Develop a reusable architecture and Open Source system software
  - The community is too small for 3 different system software models
  - The community must adopt and extend the basic architecture

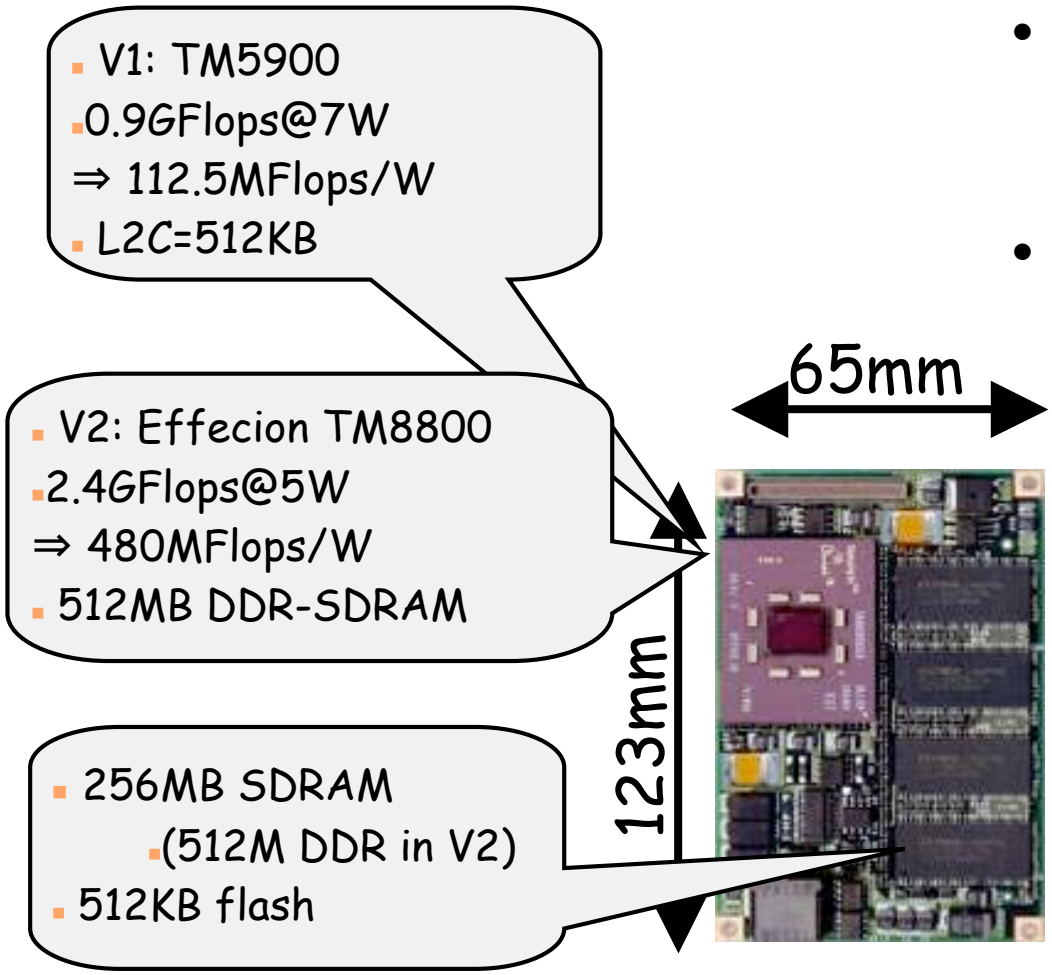
The Context for System Software: The Evolving Architecture

- Large Flat Clusters
  - Berkeley NOW makes Top500, Solaris, 1997
- Basic architecture evolves via Open Source and Linux -- system software scales poorly
- Hierarchical Linux clusters evolve
  - Cplant, Chiba City, etc
  - No adoption as target for system software devl.
- Clusters of SMPs and point solutions let systems reach 1000s of CPUs (with difficulty)
- IBM, Cray, and others develop highly scalable hierarchical platforms combining Linux, proprietary components, and Open Source HPC components
- **>>You are Here<<**
- Community embraces new platforms and develops Open Source components and extends system software capabilities

This New Model

- "What OS does BG/L run?"
  - Service Node: Linux SuSE SLES 8
  - Front End Nodes: Linux SuSE SLES 9
  - I/O Nodes: IBM-created Embedded Linux
  - Compute Nodes: Home-brew OS
- "What OS does Red Storm run?"
  - Service Nodes: Linux
  - RAS Nodes: Linux
  - I/O Nodes: Linux
  - Compute Nodes: Home-brew OS
- Extremely large systems run an "OS Suite"
  - **Functional Decomposition** trend lends itself toward a customized, optimized point-solution OS
  - **Hierarchical Organization** requires software to manage topology, call forwarding, and collective operations

JST-CREST "Megascale" Project

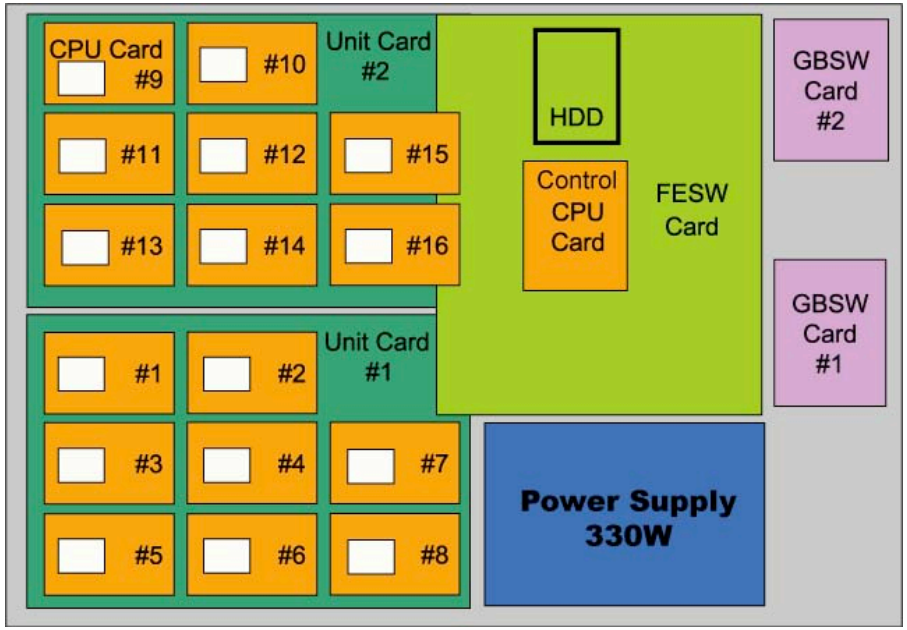


- PI: Hiroshi Nakashima (Toyohashi IT) – Megascale Cluster Federation (Grid) programming
- Co-Pis
  - Hiroshi Nakamura (U-Tokyo) – Low power processor architecture
  - Mitsuhisa Sato (Univ. Tsukuba) – Low power compiler and runtime
  - Taisuke Boku (Univ. Tsukuba) – Dependable multi-way interconnect
  - Satoshi Matsuoka (Titech) – Dependable and Autonomous Cluster Middleware

Courtesy: Satoshi Matsuoka

MegaProto Packaging (1U Chassis)

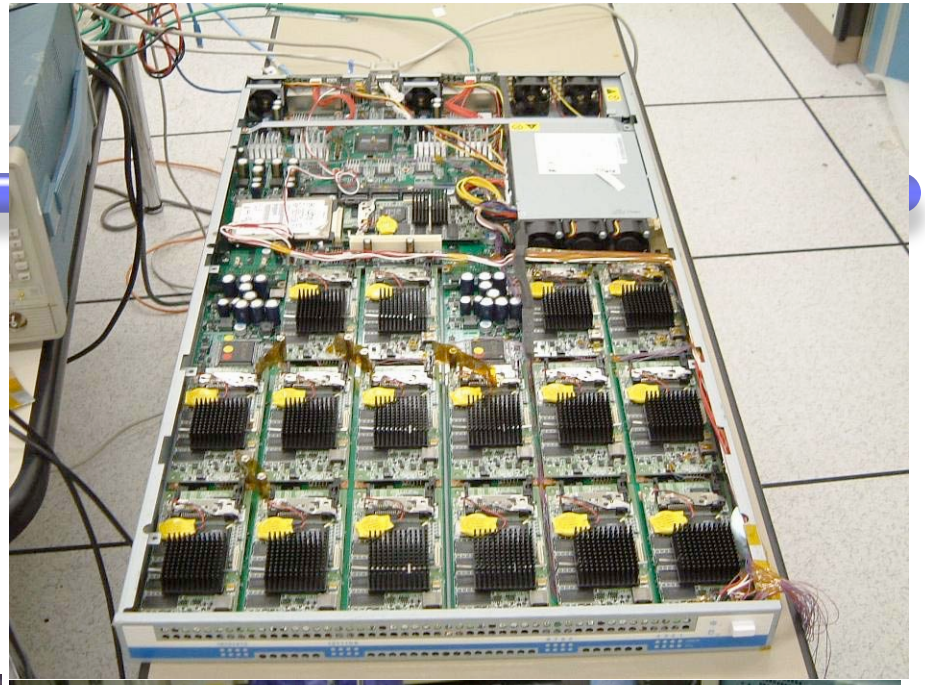
- 16+1 processor cards (fanless)
- 2 types of built-in networks on MB
  - Dual GbE, 16 x 2 (Internal) + 8 x 2(external) ports, internally switched
  - Control Network (100Base-T)
- Processor Card Upgradable (PCI-X based interface)
- Control CPU, Local HDD
- 330W Power Supply

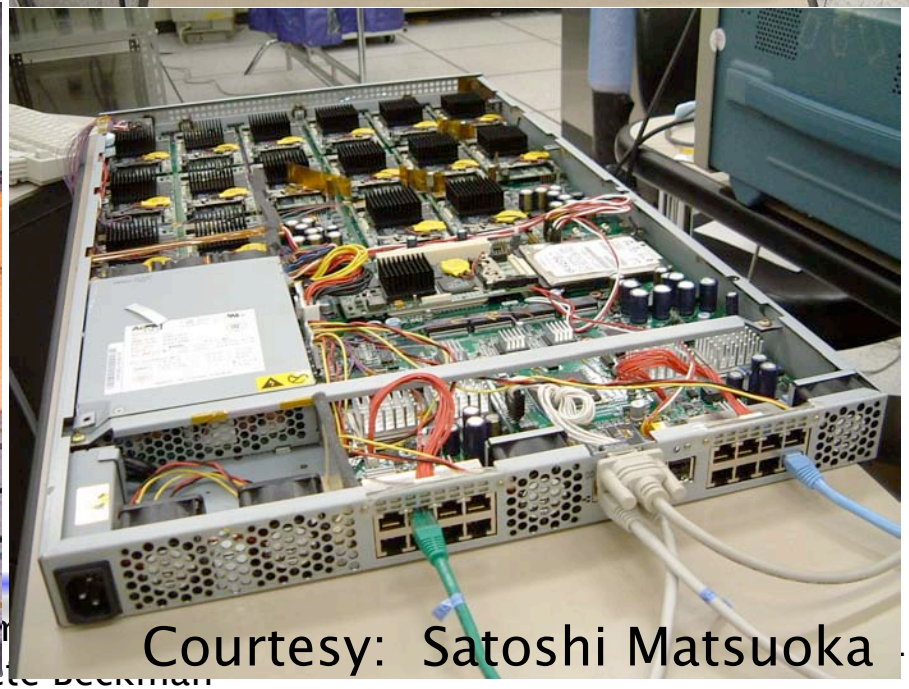


Courtesy: Satoshi Matsuoka

The Systems

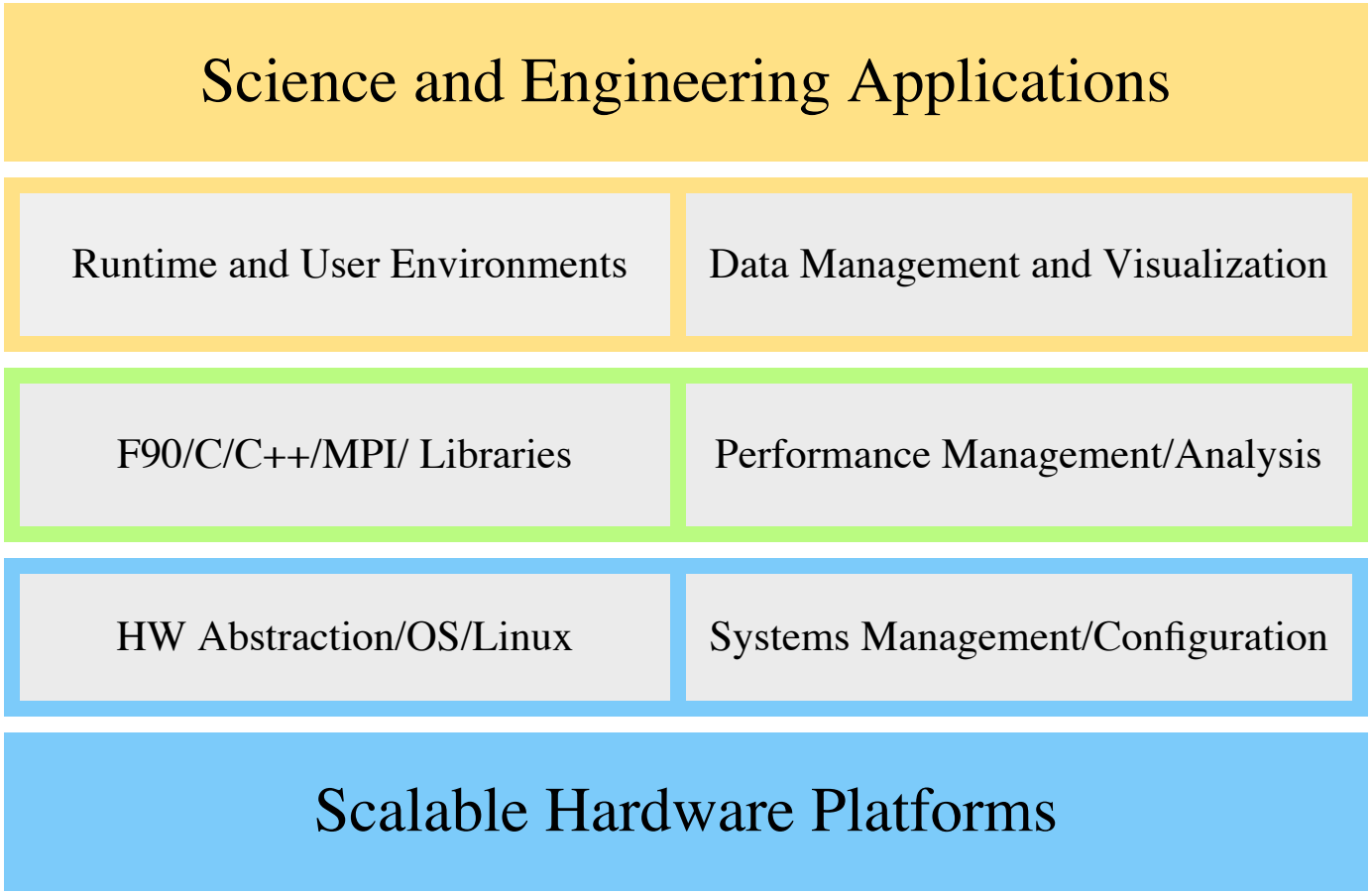
- Fabrication by IBM Japan (same fab team as BlueGene/L)
- Initial 2 Units (32 PE) delivered, being tested
- 32 proc Linux Cluster





Courtesy: Satoshi Matsuoka

Our Challenge:
Optimize The SW Stack For New Architecture Class



Including:

- Parallel File Systems
- High Performance MPI-2 & MPI-IO
- Global Arrays, UPC, CAF
- Parallel workflow and scripting
- Job and resource management
- Pipes to real-time viz
- Fault tolerance
- Collective operating system calls
- Performance tools that work cooperatively across all components
- Improve debugging

Good News: Progress Already!

- ANL has developed a Linux I/O node toolkit that can be distributed to developers
  - Special thanks to LLNL & IBM
- What is done:
  - Linux kernel, config., compile & ramdisk tools, etc.
- How it is being used:
  - Extend capabilities of I/O node
  - Build kernel modules
  - Build performance tools (TAU extension)
- The Open Source model is working
  - The World's First Parallel Open Source File System for BG/L is now working!
- We will release the env. after we finish skiing
- More utils coming...


Goals for Workshop

- Learn about the system software currently available and in development for BG/L
- Share experiences and solutions
- Prioritize system software enhancements and requirements
- Find areas where community Open Source development can extend the BG/L environment
- Organize community efforts and build collaboration